

Moon-1

Data Path

CS31

Pascal Van Hentenryck



Overview

Moon-1 (II)

- Single Cycle Implementation of Moon
- Data-Path

What is a program?

```
zero_c:      JZ      zero_c
endif:      LIM     1
            SOU
            LIM     0
            JZ      endif
            SOU
            ...
```

Binary	Hexa	Assembly
11000101	C5	JZ zero_c
01000001	41	LIM 1
10000000	80	SOU
01000000	40	LIM 0
11000110	C6	JZ endif
10000000	80	zero_c: SOU

Single Cycle Implementation

Key Idea

- Each instruction executes in a single clock cycle

Advantages

- Simplicity of the data-path
- Simplicity of the control

Inconveniences

- See later

Moon-1

Basic Cycle of Execution

- Fetch and Decode
- Execute

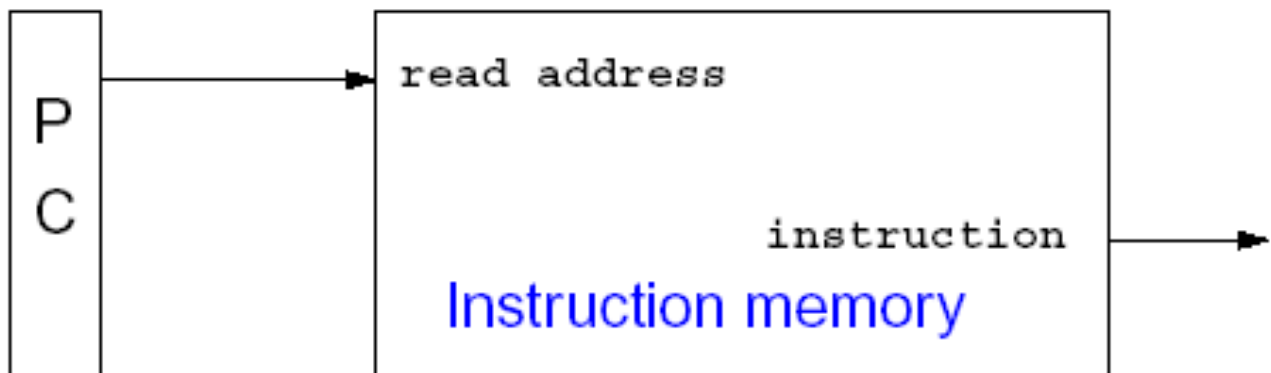
Fetch

Where is the program?

- Instruction Memory (ROM)

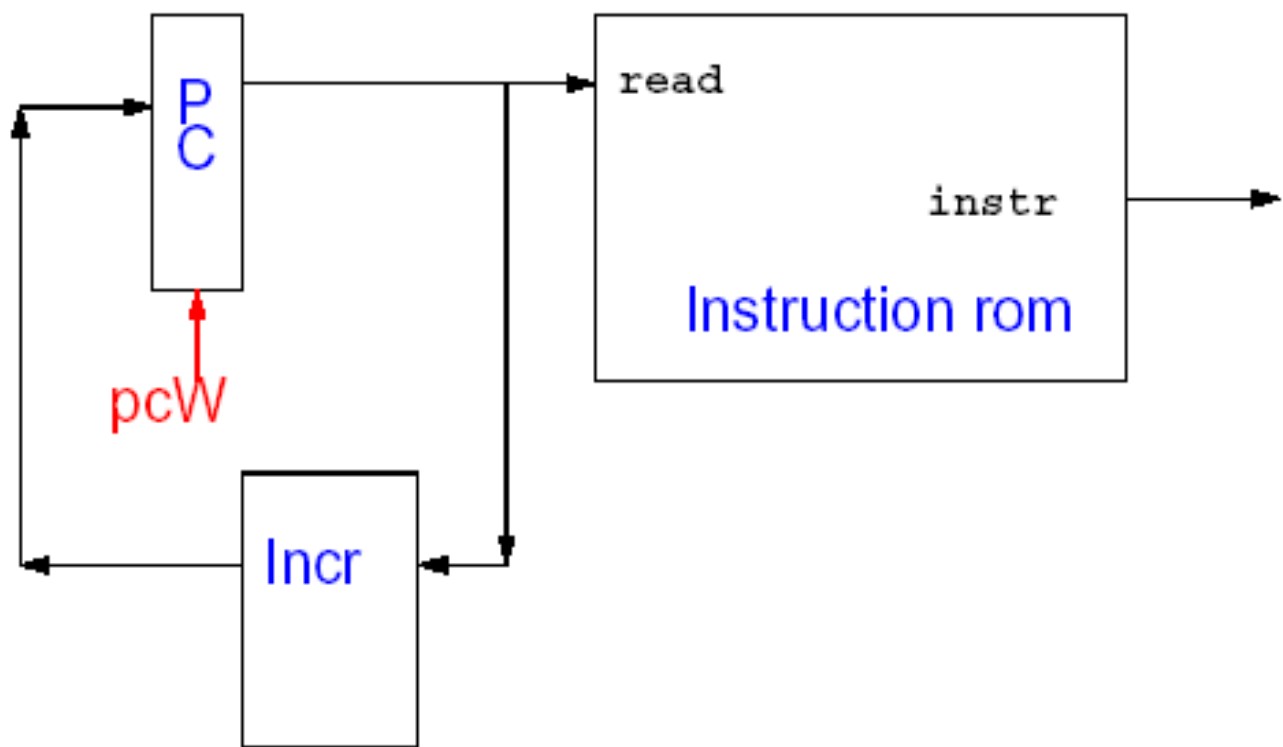
Where to find the next instruction?

- Special register: PC (program counter)



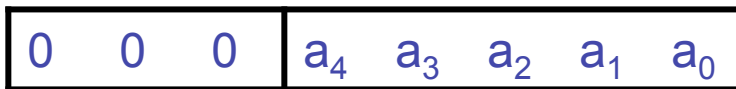
Preparing for next instruction

Need to increment the PC by one



Opcodes: Instruction Format

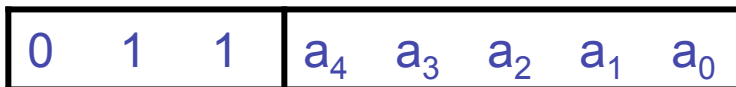
LAD addr



LIM value



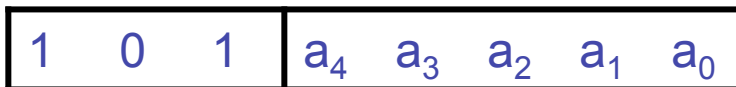
SAD addr



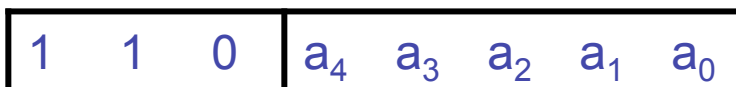
SOU



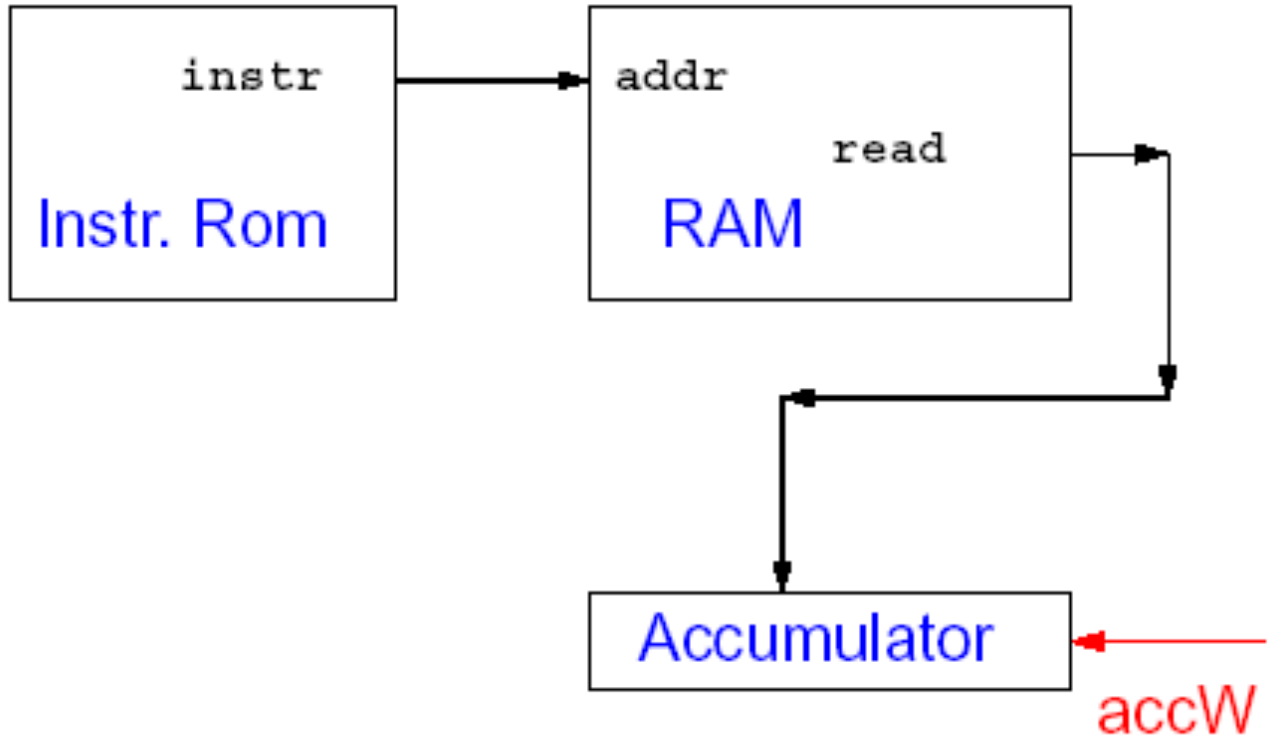
SUB addr



JZ addr

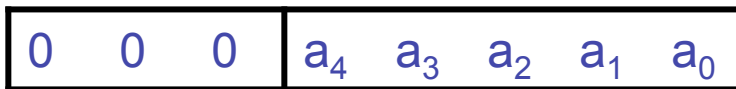


Load Instruction



Opcodes: Instruction Format

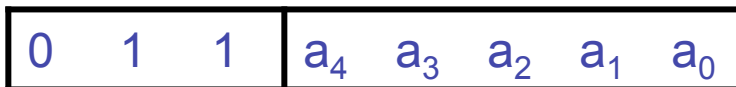
LAD addr



LIM value



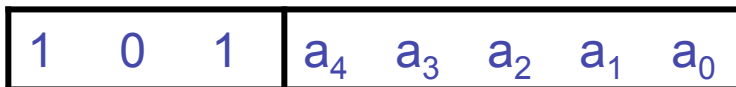
SAD addr



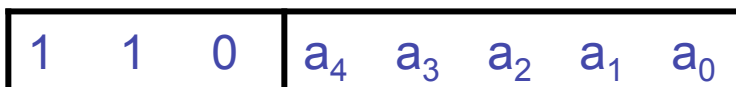
SOU



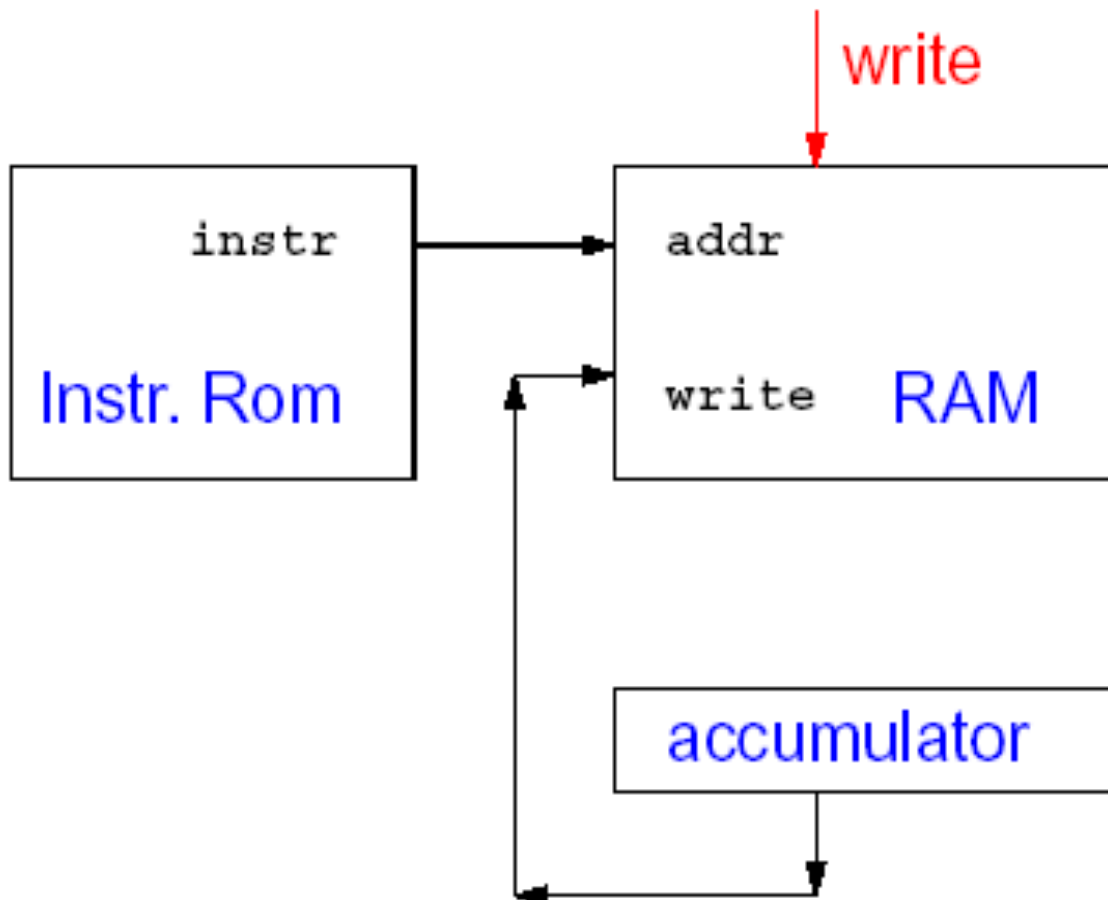
SUB addr



JZ addr

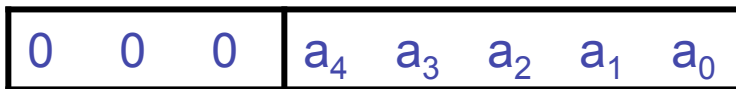


Store Instruction



Opcodes: Instruction Format

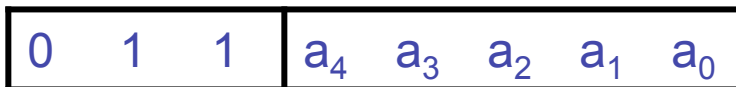
LAD addr



LIM value



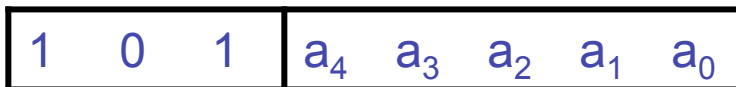
SAD addr



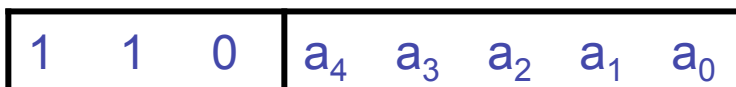
SOU



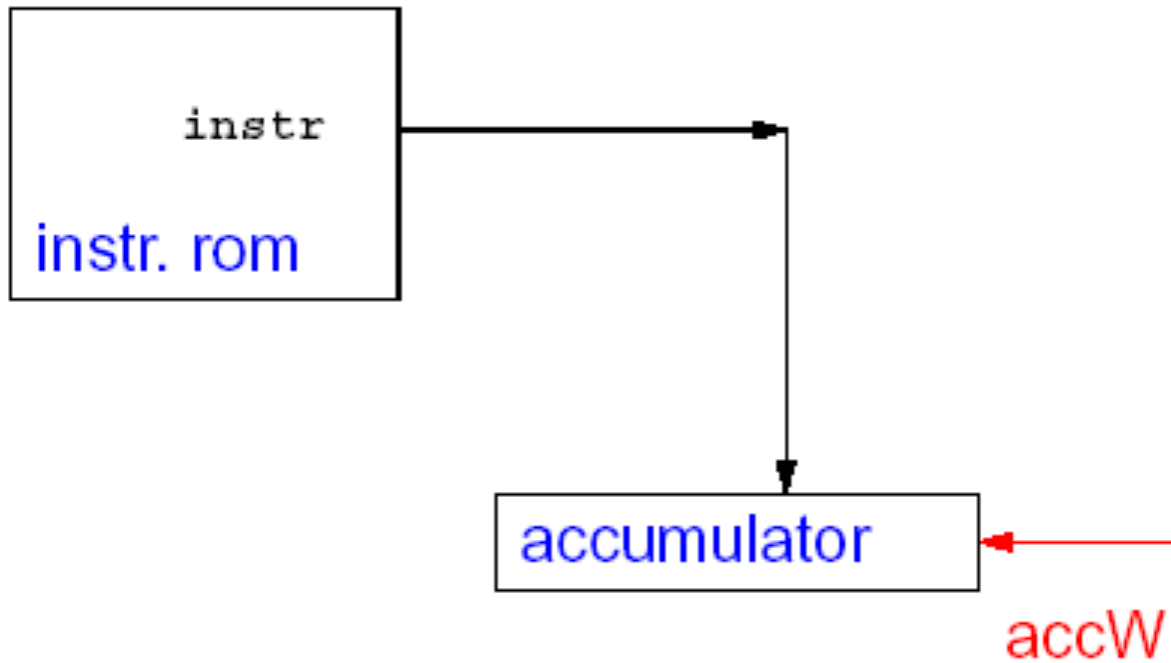
SUB addr



JZ addr

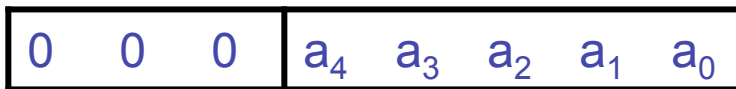


LIM Instruction



Opcodes: Instruction Format

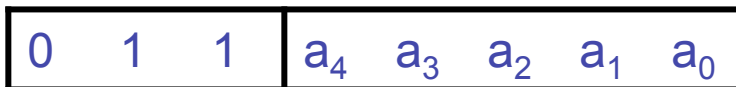
LAD addr



LIM value



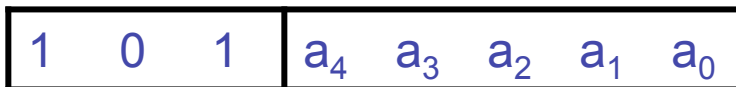
SAD addr



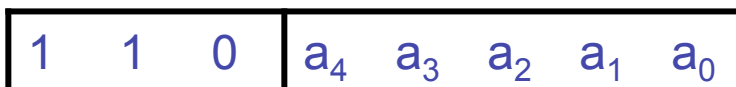
SOU



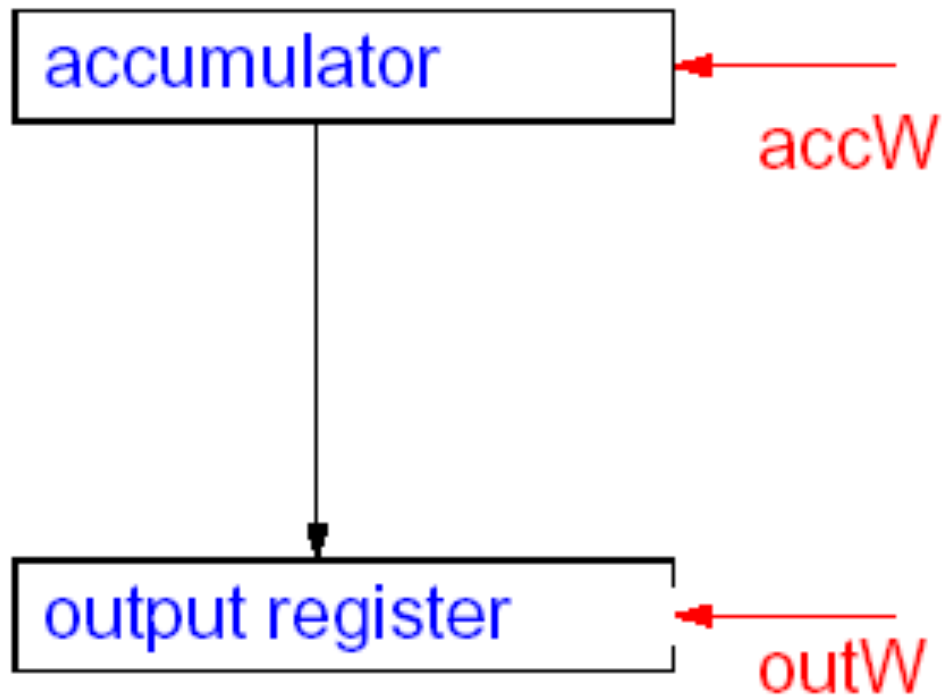
SUB addr



JZ addr

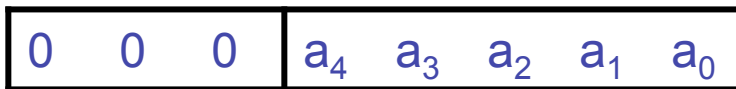


SOU Instruction



Opcodes: Instruction Format

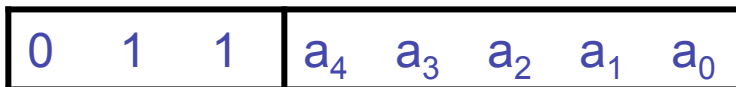
LAD addr



LIM value



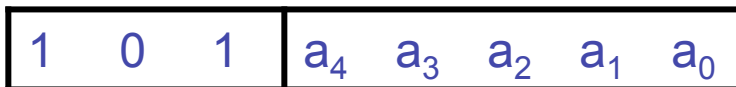
SAD addr



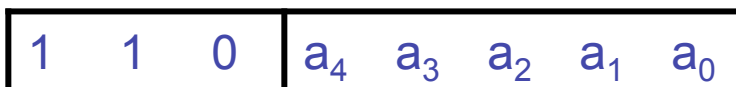
SOU



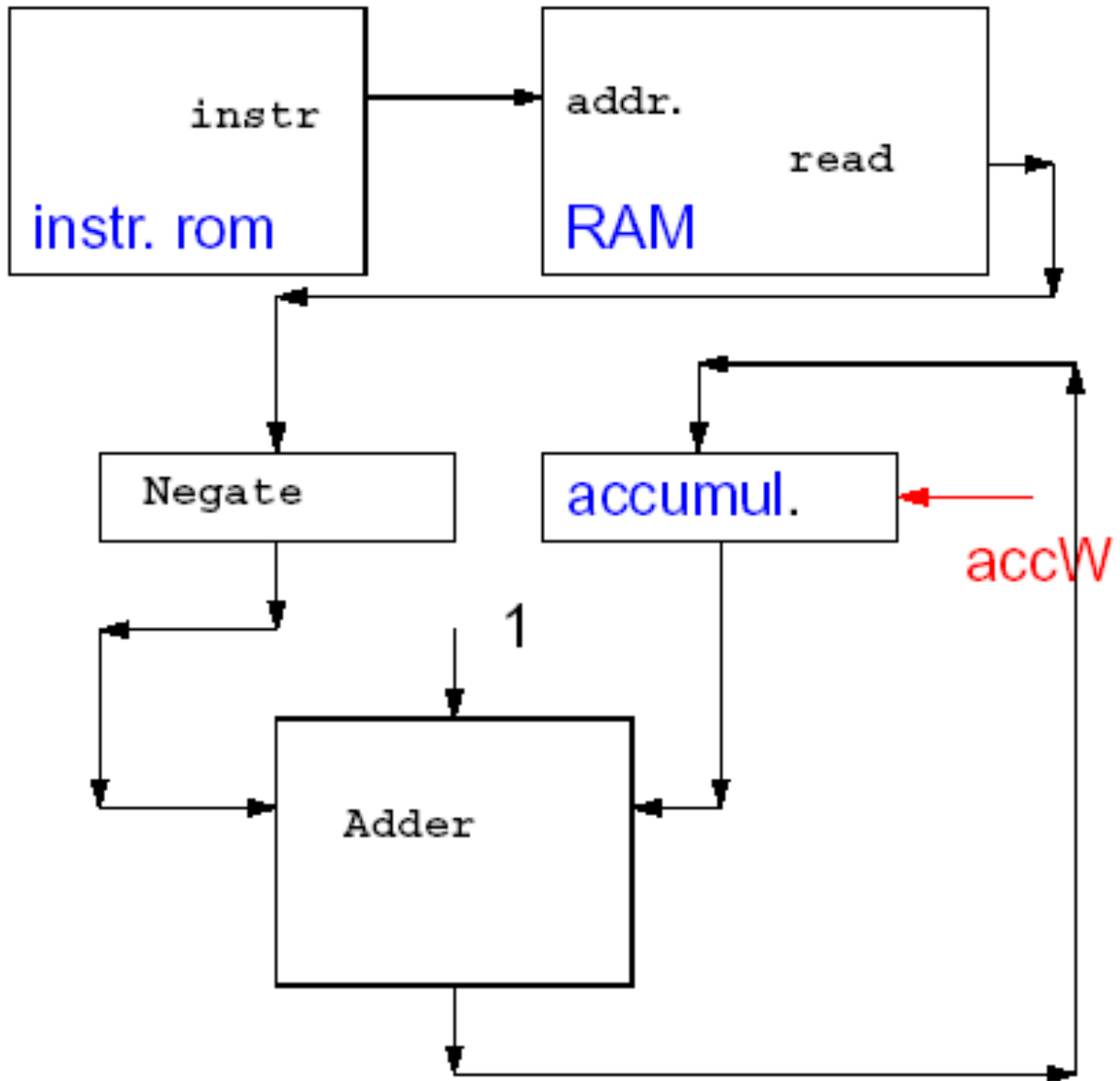
SUB addr



JZ addr

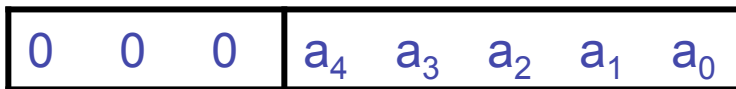


SUB Instruction



Opcodes: Instruction Format

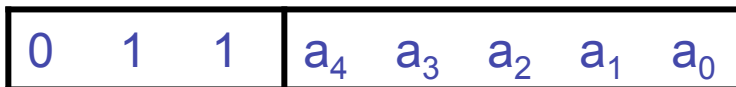
LAD addr



LIM value



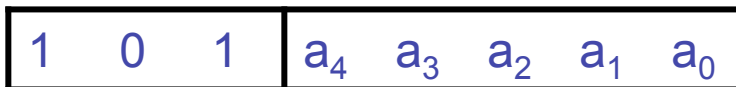
SAD addr



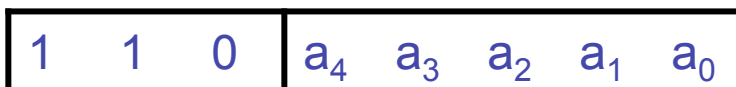
SOU



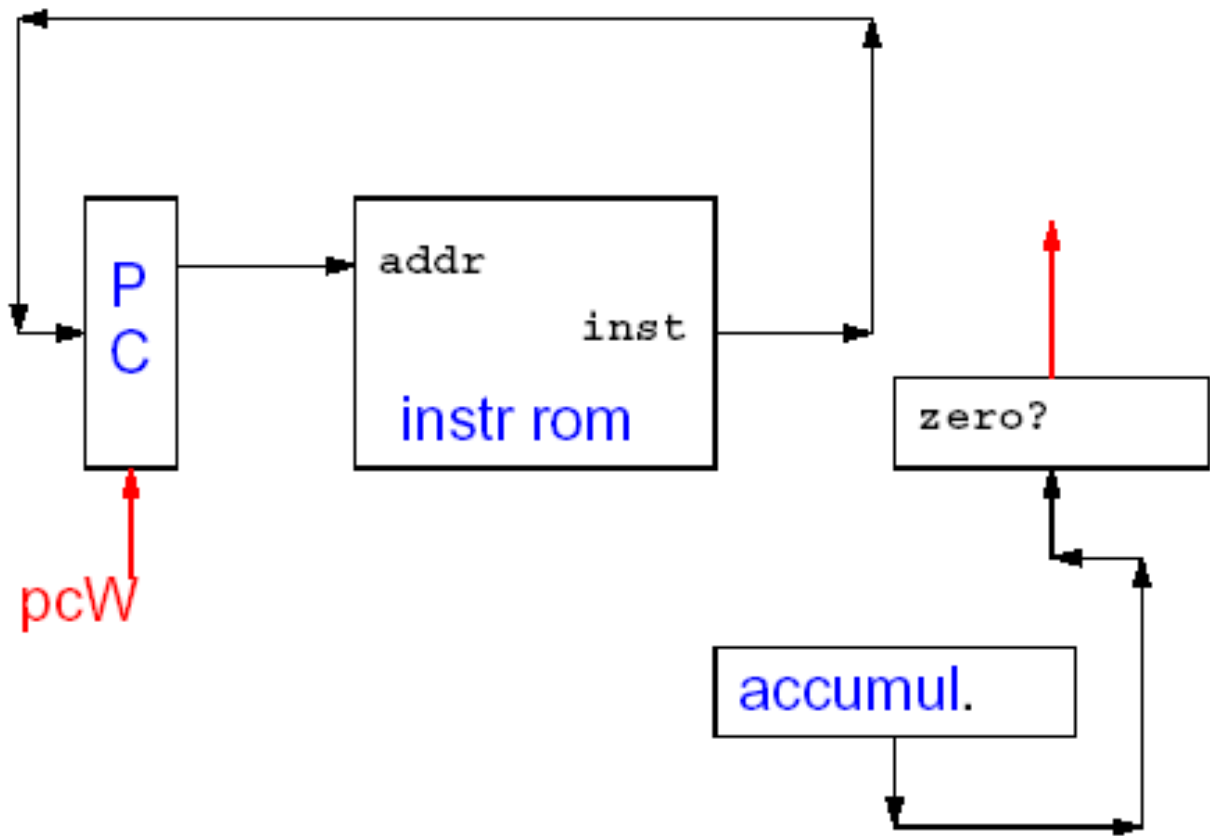
SUB addr



JZ addr



Jump Instruction



Give Piece a chance

How to put the pieces together?

- Need some multiplexers

PC

- all instructions but jump / jump

Accumulator

- LIM / SUB / LAD

